

Atividade 1

RMarkdown

Carlos Ronchi

19 de Maio de 2017

Primeira atividade avaliativa do curso de RMarkdown

Este trabalho tem como objetivo a aplicação de comandos para o *RMarkdown*.

Equações diferenciais no R.

Vamos resolver algumas equações diferenciais ordinárias utilizando o **R**.

Modelo de Lorentz

Vamos considerar o modelo de Lorenz, que é um sistema de equações diferenciais dado por

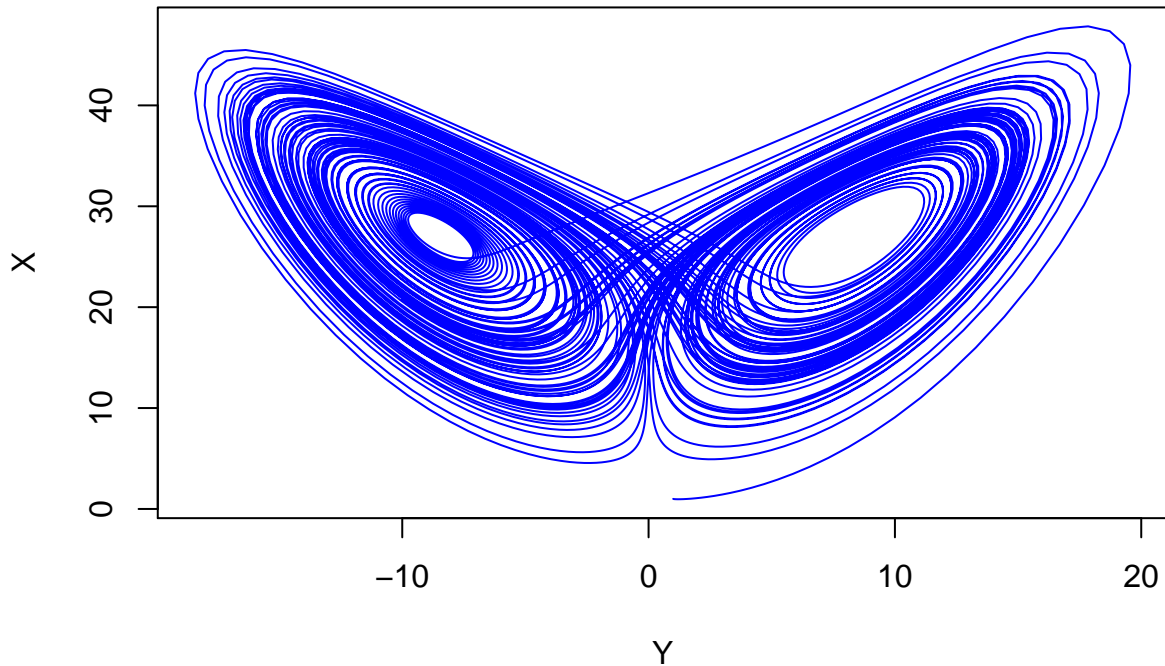
$$\begin{aligned}\frac{dx}{dt} &= a(y - x) \\ \frac{dy}{dt} &= x(b - z) - y \\ \frac{dz}{dt} &= xy - cz.\end{aligned}\tag{1}$$

```
library(deSolve)
a <- -8/3 ; b <- -10 ; c <- 28
yini <- c(X = 1, Y = 1, Z = 1)
Lorenz <- function (t, y, parms) {
  with(as.list(y), {
    dX <- a*X + Y*Z
    dY <- b*(Y-Z)
    dZ <- -X*Y + c*Y - Z
    list(c(dX,dY,dZ))
  })
}
times <- seq(from=0, to=100, by=0.01)
out <- ode(y = yini, times = times, func = Lorenz, parms = NULL)
```

Após a modelagem da equação diferencial acima, vamos plotar a solução.

```
plot(out[, "Y"], out[, "X"], type = "l", xlab = "Y", ylab = "X", main = "Solução Borboleta", col="blue")
```

Solução Borboleta



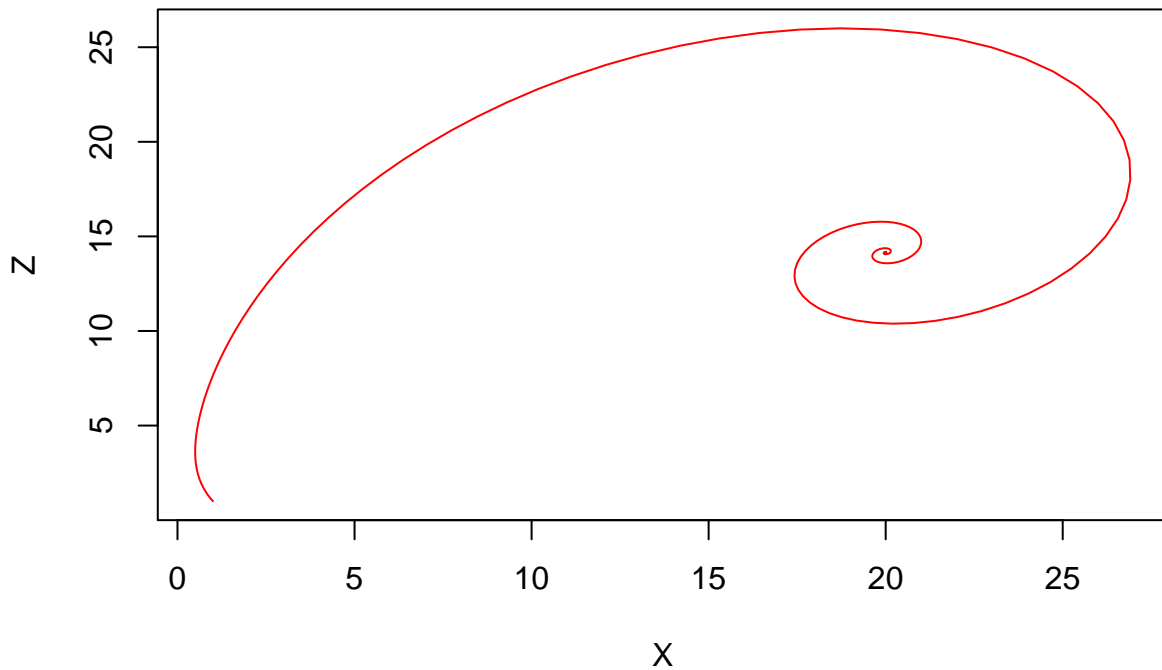
O modelo de Lorenz também é conhecido como Atrator de Lorenz, devido ao comportamento das soluções. Vamos plotar mais algumas soluções para a , b e c distintos.

```
a <- -10 ; b <- -8/3 ; c <- 21
Lorenz <- function (t, y, parms) {
  with(as.list(y), {
    dX <- a*X + Y*Z
    dY <- b*(Y-Z)
    dZ <- -X*Y + c*Y - Z
    list(c(dX,dY,dZ))
  })
}
times <- seq(from=0, to=1000, by=0.01)
out <- ode(y = yini, times = times, func = Lorenz, parms = NULL)
```

Sendo o seu gráfico dado abaixo.

```
plot(out[, "X"], out[, "Z"], type = "l", xlab = "X",
      ylab = "Z", main = "Solução Fibonacci", col="red")
```

Solução Fibonacci

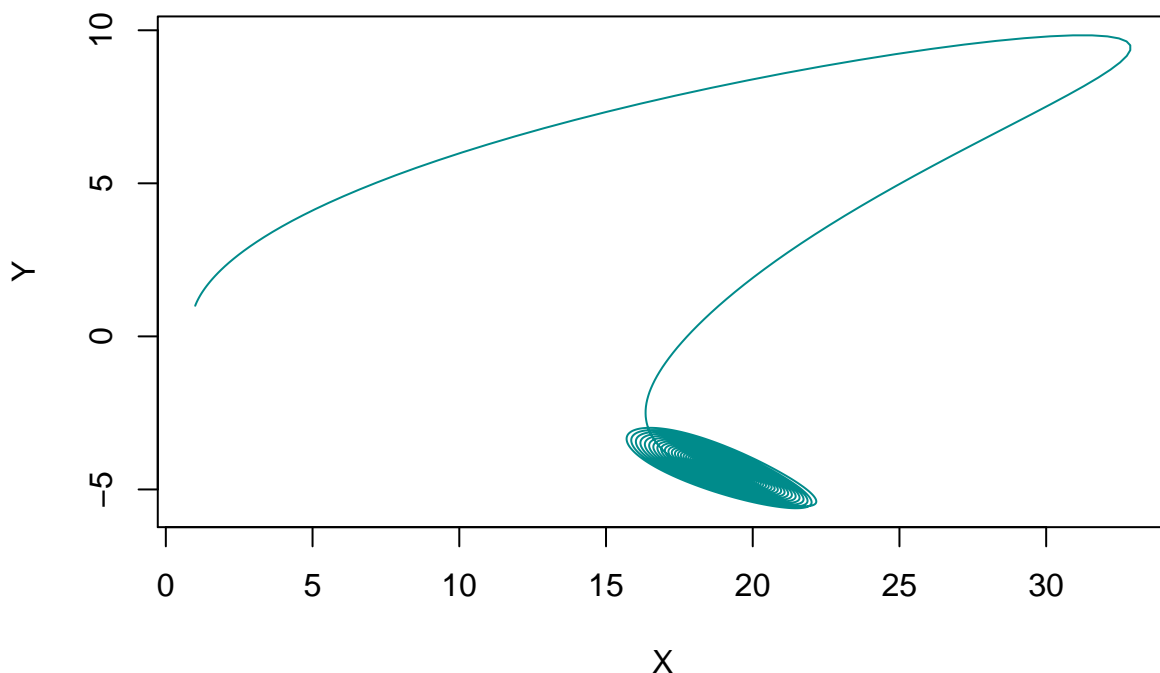


E para outros parâmetros a , b e c , temos que

```
a <- -1 ; b <- -8/3 ; c <- 20
Lorenz <- function (t, y, parms) {
  with(as.list(y), {
    dX <- a*X + Y*Z
    dY <- b*(Y-Z)
    dZ <- -X*Y + c*Y - Z
    list(c(dX,dY,dZ))
  })
}
times <- seq(from=0, to=100, by=0.01)
out <- ode(y = yini, times = times, func = Lorenz, parms = NULL)

plot(out[, "X"], out[, "Y"], type = "l", xlab = "X",
      ylab = "Y", main = "Solução Redemoinho", col="cyan4")
```

Solução Redemoinho



Vamos agora gerar um gif para o último atrator de Lorenz.

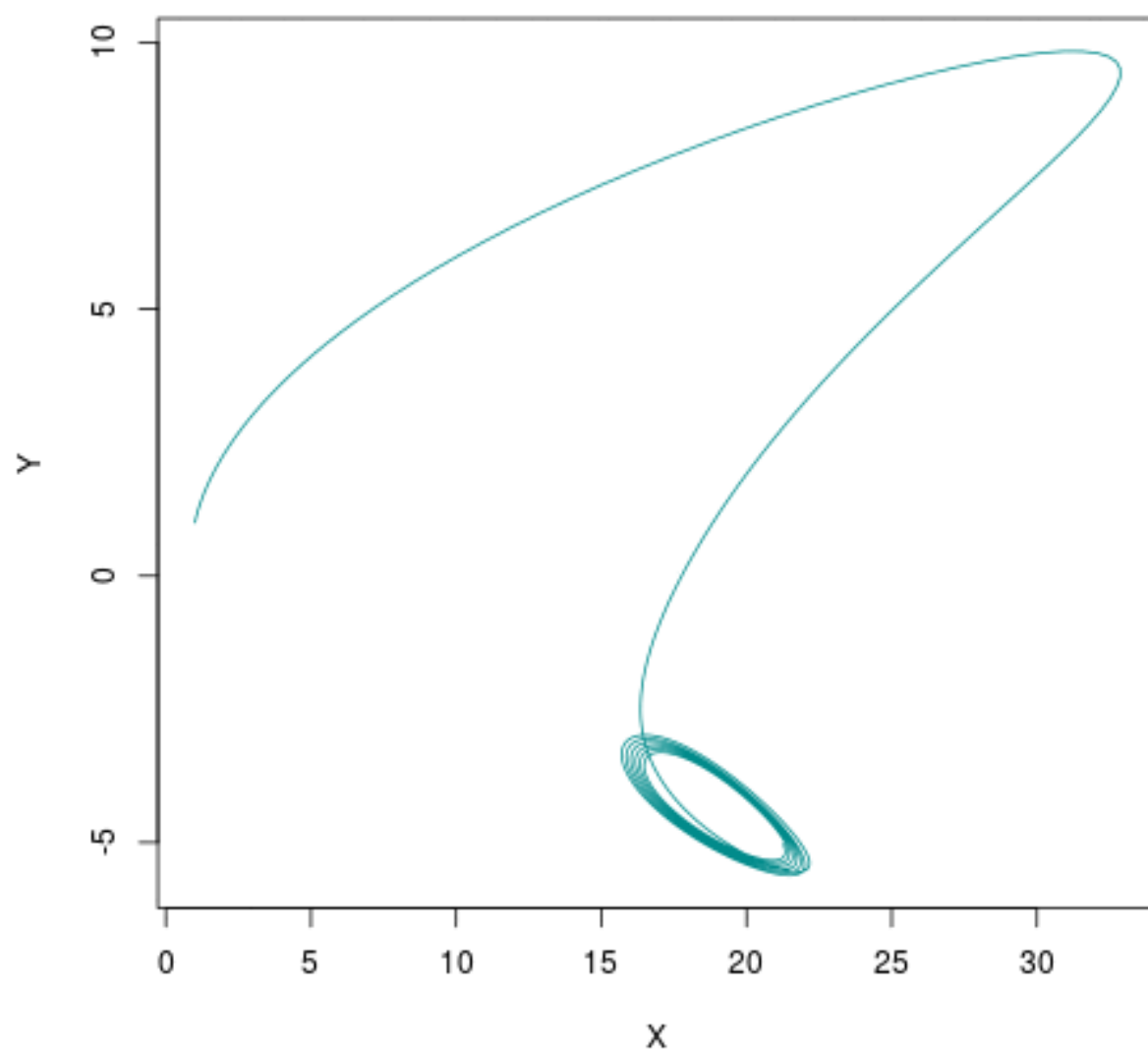
```
frames = 20
for(i in 1:frames){
  a <- -1 ; b <- -8/3 ; c <- 20
  Lorenz <- function (t, y, parms) {
    with(as.list(y), {
      dX <- a*X + Y*Z
      dY <- b*(Y-Z)
      dZ <- -X*Y + c*Y - Z
      list(c(dX,dY,dZ))
    })
  }
  times <- seq(from=0, to=i, by=0.01)
  out <- ode(y = yini, times = times, func = Lorenz, parms = NULL)
  name = paste('0',i,"plot.png",sep='')
  png(name)
  plot(out[, "X"], out[, "Y"], type = "l", xlab = "X",
        ylab = "Y", main = "Solução Redemoinho", col="cyan4")
  dev.off()
}
```

Acima geramos as imagens para criar um *gif*. Para juntar as imagens, temos que usar o terminal. Para tanto, basta entrar no diretório onde as imagens estão salvas e rodar o seguinte código

```
convert *.png -delay 3 -loop 0 redemoinho.gif
```

Assim, temos o seguinte *gif*.

Solução Redemoinho



Com isso, encerramos a primeira Atividade do curso de RMarkdown.